Joshua Bryner
jbryner@cc.usu.edu
Home: 753-0413

Monday April 28, 2008

YangQuan Chen
Department of Electrical Engineering
Utah State University

Dear Dr. Chen:

The following document contains my Final Design Report. I had a great educational experience through the design and implementation phases of this project. Much has been learned about sensors and controlling different elements based on inputs. Also the design, programming, control and interfacing a microcontroller to these elements has been a challenge in itself. This report will discuss the overall approach I have taken to complete the project and the process of implementing that approach. I will be happy to answer any questions and concerns you may have concerning my project. I am extremely excited to be able to display this project during the last week of classes to fellow students, and other faculty and staff that may be in attendance.

Thank you,

Joshua Bryner

Senior Project Final Design Report for

Automatic Sprinkler System

ECE 4850 Design III

April 28th, 2008

Joshua Bryner

Instructor Approval _____ /_____

Paul Wheeler Date

Electrical and Computer Engineering Dept.

Utah State University

Instructor Approval _____/_____

YangQuan Chen Date

Department of Electrical and Computer Engineering

Utah State University

Table of Contents

TABLES ..... **II**

ABSTRACT ..... **- 1 -**

1. INTRODUCTION ..... **- 1 -**

1.1 Purpose ..... **- 1 -**
1.2 Identified Need ..... **- 2 -**
1.3 Problem Definition ..... **- 2 -**
1.4 Objectives ..... **- 3 -**
1.5 Deliverables ..... **- 4 -**

2. CONCEPTUAL DESIGN ..... **- 5 -**

2.1 Problem Review ..... **- 5 -**
2.2 Summary of Specifications ..... **- 5 -**
2.3 Solution Alternatives ..... **- 6 -**

3. ENGINEERING APPROACH ..... **- 6 -**

3.1 Block Diagram ..... **- 6 -**
3.2 Moisture Sensor Subsystem ..... **- 7 -**
3.2.1 Moisture Sensor ..... **- 7 -**
3.2.2 Analog Multiplexer ..... **- 8 -**
3.2.3 A/D Converter ..... **- 9 -**
3.3 Sprinkler Valve Subsystem ..... **- 12 -**
3.3.1 Digital Multiplexer ..... **- 12 -**
3.3.2 24 VAC Switching ..... **- 13 -**
3.4 Graphical User Interface Subsystem ..... **- 15 -**
3.4.1 The Design Process of the GUI ..... **- 15 -**
3.4.2 Graphical User Interface Flow ..... **- 19 -**
3.5 Microcontroller ..... **- 20 -**
3.5.1 Characteristics ..... **- 20 -**
3.5.2 Software Flow ..... **- 23 -**
3.5.3 Demo Program ..... **- 24 -**
3.5.4 I/O Ports ..... **- 25 -**

4. BUDGET ..... **- 26 -**

5. PROJECT MANAGEMENT ..... **- 27 -**

5.1 Schedule Tasks ..... **- 27 -**

6. CONCLUSION ..... **- 28 -**

Figures

FIGURE 1 BLOCK DIAGRAM ..... **- 6 -**
FIGURE 2 ECHO EC-5 SENSOR ..... **- 7 -**
FIGURE 3 EC-5 SPECIFICATIONS ..... **- 8 -**
FIGURE 4 MAX396 PIN OUT ..... **- 9 -**
FIGURE 5 MAX396 MAXIMUM RATINGS ..... **- 9 -**
FIGURE 6 ADC0820 PIN OUT ..... **- 10 -**
FIGURE 7 SENSOR POWER CIRCUIT ..... **- 11 -**
FIGURE 8 SENSOR DATA CIRCUIT ..... **- 11 -**
FIGURE 9 SPRINKLER VALVE CONTROL CIRCUIT ..... **- 12 -**
FIGURE 10 MAX358 PIN OUT ..... **- 13 -**
FIGURE 11 MAX 358 ABSOLUTE MAXIMUMS ..... **- 13 -**

FIGURE 12 MOC3011 TRIAC DRIVER ..... - 14 -
FIGURE 13 TRIAC AND TRIAC DATA ..... - 14 -
FIGURE 14 CONNECTION SCREEN ..... - 17 -
FIGURE 15 LOGIN SCREEN ..... - 17 -
FIGURE 16 EDIT USERS SCREEN ..... - 17 -
FIGURE 17 MAIN SCREEN ..... - 19 -
FIGURE 18 GUI FLOW DIAGRAM ..... - 20 -
FIGURE 19 ZILOG EZ80F91 ..... - 21 -
FIGURE 20 DEVELOPMENT BOARD HEADER ..... - 22 -
FIGURE 21 DEVELOPMENT BOARD HEADER 2 ..... - 22 -
FIGURE 22 SOFTWARE FLOW ..... - 23 -
FIGURE 23 DEMO SOFTWARE FLOW ..... - 24 -

Tables

TABLE 1 I/O PORTS ..... - 26 -
TABLE 2 PROJECT COST ..... - 27 -
TABLE 3 GANTT CHART ..... - 28 -

Abstract

Joshua Bryner, per request of the Utah State University engineering department, will design an Automatic Sprinkling system. The requirements include using moisture sensors, the controller to connect to a home network via Ethernet, and a Graphical User Interface (GUI) that can change the settings on the controller via the home network. The system must allow the connection of up to eight sprinkler zones. The system must also allow for up to sixteen moisture sensors. The system must use the smallest amount of input and output pins to interface with the zones and sensors.

The GUI software must be able to run on Windows operating systems. Also, the software must allow the user to add additional sprinkler zones at anytime. This system must allow users to add it to a current configuration of sprinkler valves that may already be in place. This report will define this project's objectives and constraints, deliverables, engineering approach, budget, and management.

1. Introduction

1.1 Purpose

When a family purchases an older home they are faced with many questions. Why did they install this? Does the lawn have a sprinkler system? Is the existing sprinkler system easy to use? These are only a fraction of the questions a person could have. When it comes to a home one of the most appealing aspects to people walking by is the lawn. This can cause more stress than is needed to fiddle around with the timer systems or even installing a new sprinkler system.

For someone with an electronic background, configuring a sprinkler system may be a simple task. However, for the rest of the population just the thought of playing with all the timers makes them want to water the lawn by hand. So people will spend a lot of money to have someone else install or repair a system, which impacts the amount of money they can spend on a sophisticated system. Once the system is installed it can still be difficult to tinker with the timers to get their lawn watered correctly.

1.2 Identified Need

The problem with sprinkler systems today is they allow users to either under water or over water their lawns. Most people set their timers to water at the wrong time, too often, or not often enough, and people don't know there is a problem until it is physically noticeable. Even if a person spoils themselves and purchases a sprinkler controller that has a moisture sensor, the controllers typically only allow for one sensor to be attached. This can cause two problems. One is if the position that the sensor is in may be moist more often and cause the rest of the lawn to dry out. The other problem is if the position is in a location that dries out quicker, and then the rest of the lawn will get over watered. So even with an expensive system lawns can still be over and under watered.

1.3 Problem Definition

Lawn care can be a very tedious job even with the use of a sprinkler system. Timers have to be adjusted monthly to account for shorter and longer days so that the sprinklers will turn on at the correct time. The user has to be

mindful of when weather patterns change and turn on or off the system so the lawn is properly watered.

The main issue with a sprinkler system is determining when the system should water the lawn. The most common method is to have a timing system to water the lawn at the same time on certain days of the week. This type of system does a fair job but can cause the problems described above. To fix this, a moisture sensor can be added to allow the system to have information if the lawn really needs water. Since current systems only allow for one sensor, a better design would be to allow a moisture sensor for each section (or zone) of the sprinkler system. Another concern is that with the changing seasons, the amount of time to water during one month is different from the next. The final concern is that the system will need to be connected to a computer through a microcontroller to control the inputs and outputs.

1.4 Objectives

The automated sprinkler system will monitor the moisture of the lawn and when the time comes to turn on the water, the system will check the moisture of the area to water and will decide if it needs to water. The system will also allow the users to change settings on the microcontroller through a personal computer (PC). A graphical user interface (GUI) will allow users to change the settings on the microcontroller.

Users will have the ability to start and stop the automated sprinkler system when ever they desire. The GUI will also allow an administrator to add more users to the system add/delete sprinkler zones and moisture sensors. All users will

have the ability to start and stop the system and also test a specific sprinkler zone or moisture sensor.

To provide some error in the sensors the system will allow the administrator to choose how many moisture sensors each zone will have. When a zone has more than one sensor attached to it then the controller will use the average between the sensors to determine if a zone needs to be watered. This way if a sensor fails to operate correctly other sensors on a zone will still be able to provide some information of the moisture level of a zone.

1.5 Deliverables

This project will consist of three deliverables: prototype, software package, and a user's manual. The prototype consists of a microcontroller, a moisture sensor and a universal sprinkler valve. The software package will include a GUI and all information for installation and using the software. The user's manual will include connection instructions for the prototype.

The installation of the Automated Sprinkler System prototype will be very simple. The prototype will be fully assembled prior to delivery. A serial cable will need to be connected between the prototype and the personal computer and two power sources will be connected to a 120 VAC outlet.

The prototype will be incomplete without the software package installed on the personal computer. The software package will basically consist of a program executable. The executable will be compatible to run on Windows 95 through XP. Once the package is installed, the software is run by double clicking on the executable.

2. Conceptual Design

2.1 Problem Review

The first problem to look at is how current sprinkler systems allow users to over water their lawn. The main reason this happens is the timer system doesn't have any input as to what the moisture level of any area of a lawn. Along with this problem the way to configure a system is complicated and very tedious.

2.2 Summary of Specifications

1. Connect a moisture sensor.

   As a suggestion from Dr. Chen I changed this specification to allow up to sixteen moisture sensors. The user can choose which zones these are connected to. Each zone also must have at least one sensor connected for it to be added to the system.

2. Control sprinkler zones.

   Added to this specification was the system must allow up to eight zones to be connected. Also, the system must be able to be integrated into a users' current sprinkler set up.

3. The system must use the least amount of input output pins.

   This was added so that the system would have the ability to add more moisture sensors or sprinkler zones using multiplexers.

4. The system connects to a GUI via Ethernet.

   After months of trying to figure out how to get the micro controller to connect to a home network I had to scrub this specification. It was changed to connecting to a GUI using a serial port.

2.3 Solution Alternatives

The system had many alternatives for each subsystem. Section three describes the overall design approach. Alternative solutions for each subsystem are also described in section three. Also, the final decisions for each subsystem are also described.

3. Engineering Approach

3.1 Block Diagram

The Automated Sprinkler System consists of three subsystems. These systems are the moisture sensor, sprinkler valve, and graphic user interface subsystems. The moisture sensor subsystem consists of the power and data circuits shown in figure 1. Figure 1 is a block diagram illustrating the relationship between the subsystems.

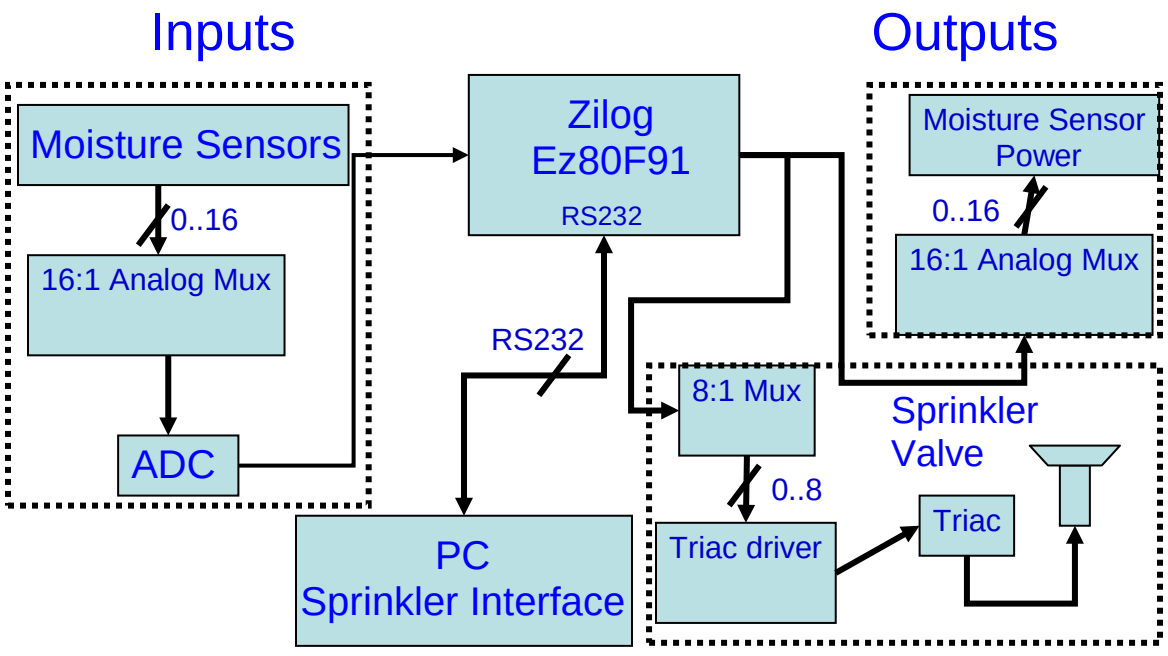


Figure 1 Block Diagram

3.2 Moisture Sensor Subsystem

When it is time for the sprinkler system to water a lawn the first thing it needs to do is turn on a moisture sensor for a zone. To do this it was necessary to design a circuit that could detect the soil moisture level. Initially, I intended to design my own circuit to do this but after further research that was a senior project in of itself. So I chose to purchase a moisture sensor from an agriculture company instead, which was very simple to integrate to a microcontroller.

Figure 2 Echo EC-5 sensor

3.2.1 Moisture Sensor

There were two choices of soil moisture sensors that were considered. The first considered was a tensiometric sensor. This type of sensor directly measures the matric potential in the soil to get a reading. This sensor was quickly decided against because of a few reasons. The first was that it has a very limited range of moisture levels so it could cause over watering. Second, the time to take a measurement could take from 30 minutes to 2 hours, which is way too long for a sprinkler system. Finally, these sensors require regular (weekly or daily) maintenance depending on the sensor and must be removed during winter.

The second was a capacitive sensor shown in Figure 2. This type of sensor determines the moisture content by measuring the capacitance

between two electrodes and then using the dielectric constant to determine the moisture. This type of sensor was chosen because first, its response time is practically instantaneous. Second, it has a high level of precision. Finally, it can be easily interfaced to a computer. Figure 3 shows the specification of the EC-5 capacitance sensor chosen.

Model	EC-5
Length	5cm
Measurement	Volumetric Water Content
Accuracy	±3% typical on all soils, up to 8 dS/m, ±1 to 2% with soil-specific calibration
Range	0–100% VWC
Electrical interface	3.5mm plug, 3-wire
Supply voltage	2.5–5V DC @ 10mA
Frequency	70 MHz
Output	Voltage, correlated linearly (soil) or polynomially (growing media) with VWC

Figure 3 EC-5 Specifications

3.2.2 Analog Multiplexer

The design requirements state to allow the system to have multiple moisture sensors. To do this while using a minimal amount of general purpose input/output pins on the microcontroller required using a 16:1 analog multiplexer. The MAX396 analog multiplexer was chosen since it has sixteen channels. Figures 4 and 5 depict the information for the multiplexer.

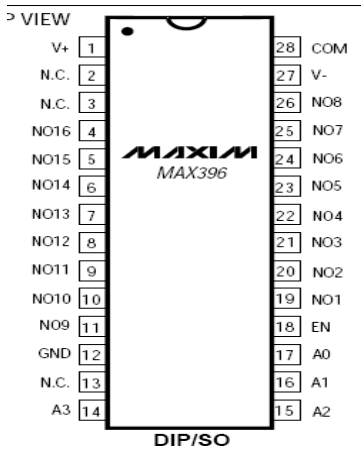


Figure 4 MAX396 Pin out

ABSOLUTE MAXIMUM RATINGS

(Voltage referenced to GND, unless otherwise noted.)
V+ .....-0.3V to +17V
V- .....+0.3V to -17V
V+ to V-.....-0.3V to +17V
Voltage into Any Terminal (Note 1).....(V- - 2V) to (V+ + 2V)
or 30mA (whichever occurs first)
Current into Any Terminal.....±30mA
Peak Current into Any Terminal.....±50mA
Continuous Power Dissipation (T_A = +70°C)
Plastic DIP (derate 14.29mW/°C above +70°C) 1143mW
Wide SO (derate 12.50mW/°C above +70°C).....1000mW
SSOP (derate 9.52mW/°C above +70°C) 762mW
PLCC (derate 10.53mW/°C above +70°C) 842mW
CERDIP (derate 16.67mW/°C above +70°C).....1333mW
Operating Temperature Ranges
MAX39_C_I .....0°C to +70°C
MAX39_E_I .....-40°C to +85°C
MAX39_MJI.....-55°C to +125°C
Storage Temperature Range.....-65°C to +150°C
Lead Temperature (soldering, 10sec) .....+300°C

Figure 5 MAX396 Maximum Ratings

3.2.3 A/D Converter

The output of the moisture sensor is an analog signal, and to allow the microcontroller to use the signal an external A/D converter is needed. Two choices of A/D converters were considered a parallel and a serial A/D converter. To minimize pin usage the serial A/D converter was chosen in the beginning. This design was later terminated after research of creating a serial interface with the general input/output proved that option prohibitive. Creating a serial interface with the microcontroller would have required a lot of timers and a lot of interrupt controlling in software.

The parallel A/D converter was then chosen to be added to the design. This type of converter did take more general input/output pins but was easier to interface to the microcontroller. The ADC0820 is an eight bit parallel A/D converter that uses an external voltage reference to decode the signal. For this system, since the moisture sensors send out a level from .5V to 2V at 5V, I would need a reference of +5V and 0V. To get the number that the A/D converter outputs this equation was calculated: DigatalNum = Output*(V_{ref}(+)-V_{ref}(-))/256. With the 5V V_{ref}(+) and a 0V V_{ref}(-), the equation becomes DigitalNum = output*20 mV. Then with this given digital Num the volumetric water content can be calculated with this equation vwc = (DigitalNum*3)/20 – 25.5. Figure 6 shows the pin out of the ADC0820.

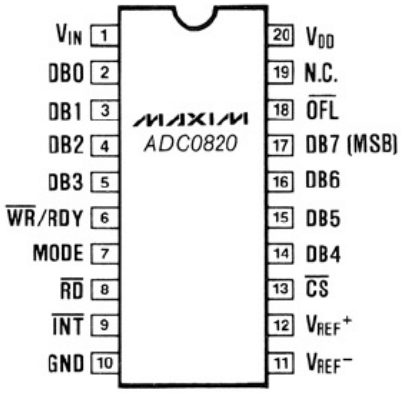


Figure 6 ADC0820 Pin out

The interface to the microcontroller is depicted in Figures 7 and 8. The EC-5 has three connections: power, data, and ground. The ground is connected to the microcontroller ground. The power signal is connected through a multiplexer to allow for more sensors than the microcontroller I/O could handle. The power is connected to the 3.33 VDC power of the

microcontroller since the I/O pins can not handle the 10mA input the sensors need.

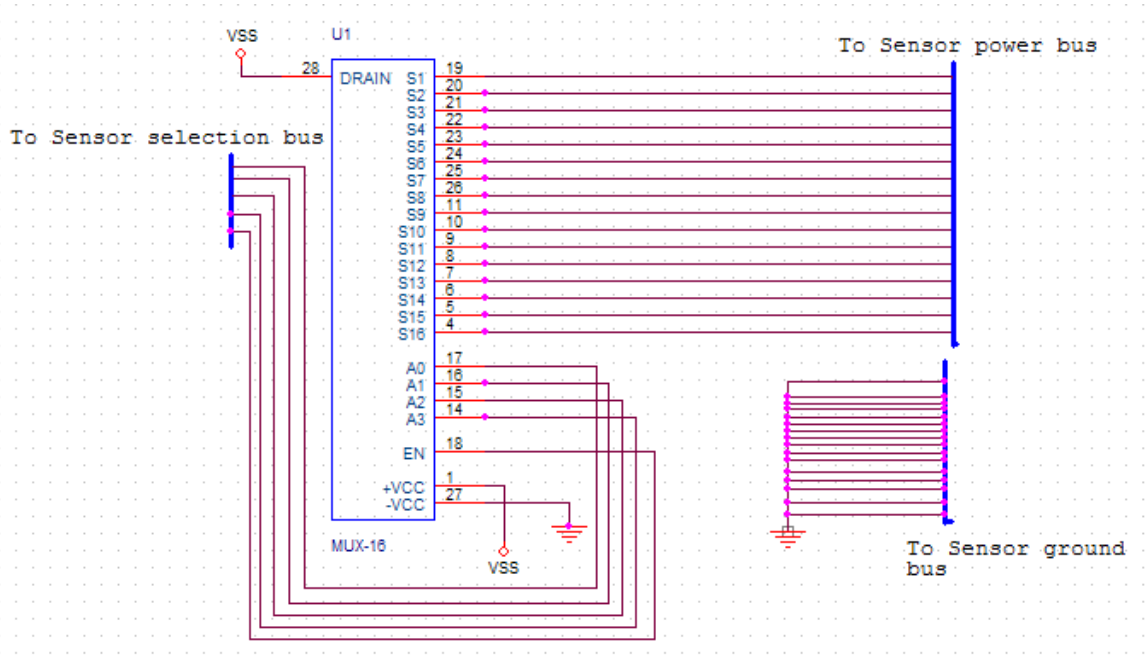


Figure 7 Sensor Power Circuit

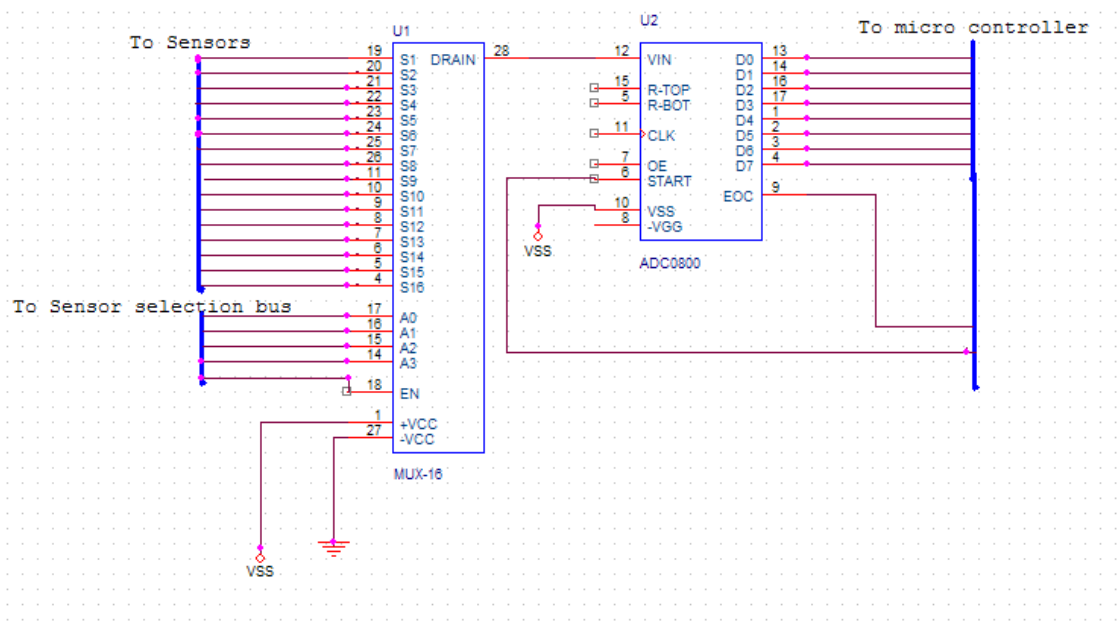


Figure 8 Sensor Data Circuit

3.3 Sprinkler Valve Subsystem

To design the sprinkler valve subsystem I didn't have very many choices to make. It seems the control of the sprinkler valves was standardized to use a 24VAC solenoid at 400 mA. The main design challenge for this subsystem was how to interface it to the microcontroller. The design specifications required up to eight zones to be able to be used so an 8:1 multiplexer was used to switch between zones. An optical isolator was chosen to separate the microcontroller from the 24 VAC power supply. To turn this power supply on and off, a triac is needed. The sprinkler valve control circuit can be seen in figure 9.

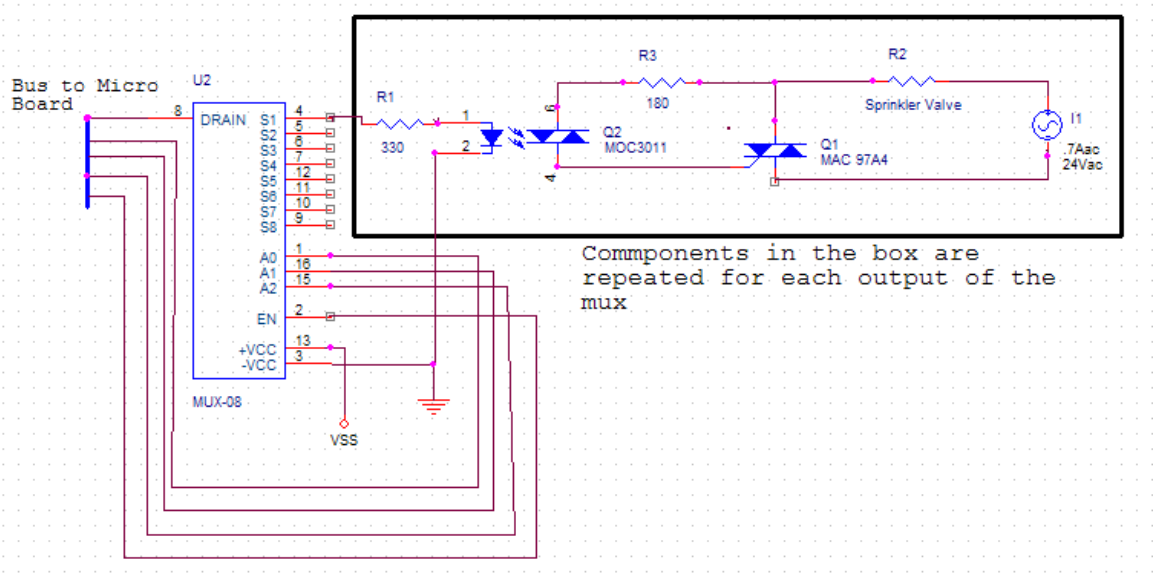


Figure 9 Sprinkler Valve Control Circuit

3.3.1 Digital Multiplexer

The design requirements state that the system must allow for up to eight sprinkler zones to be controlled. To do this while using a minimal amount of general purpose input/output pins on the microcontroller was to use an 8:1 multiplexer. The MAX358 multiplexer was chosen since it has

eight channels. Figures 10 and 11 depict the information for the multiplexer.

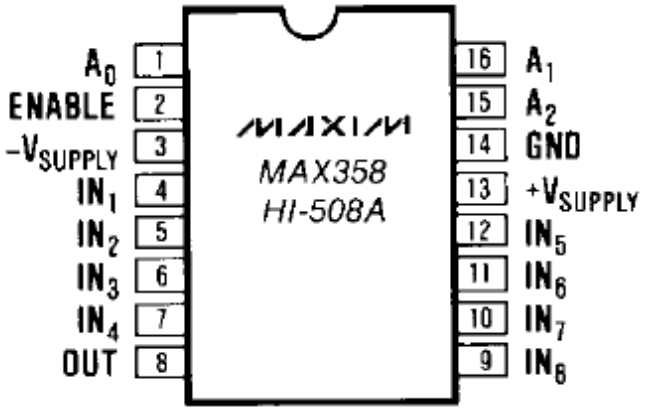


Figure 10 Max358 Pin out

ABSOLUTE MAXIMUM RATINGS

Voltage between Supply Pins ..... +44V
V^+ ..... +22V
V^- ..... −22V
Digital Input Overvoltage:
V_{EN}, V_A { $V_{Supply}(+)$ ..... +4V
{ $V_{Supply}(-)$ ..... −4V
Analog Input Overvoltage with Multiplexer Power On:
V_S { $V_{Supply}(+)$ ..... +20V
{ $V_{Supply}(-)$ ..... −20V
Analog Input Overvoltage with Multiplexer Power Off:
V_S { $V_{Supply}(+)$ ..... +35V
{ $V_{Supply}(-)$ ..... −35V

Continuous Current, S or D 20mA
Peak Current, S or D
(Pulsed at 1ms, 10% duty cycle max) 40mA
Power Dissipation (Note 1) (CERDIP) 1.28W
Operating Temperature Range:
MAX358/359M; HI-508A/509A-2, -8 ..... −55°C to +125°C
MAX358/359C; HI-508A/509A-5 ..... 0°C to +75°C
MAX358/359E ..... −40°C to +85°C
Storage Temperature Range ..... −65°C to +150°C

Note 1: Derate 12.8mW/°C above T_A = +75°C

Figure 11 Max 358 Absolute Maximums

3.3.2 24 VAC Switching

A great deal of understanding was gained about triacs and how they function during the design of the sprinkler valve control circuit. During testing and implementation it was learned that it was necessary to send a continuous pulse to the triac in order to keep it switched on. A triac is designed to allow electricity to flow whenever the voltage rises above the gate trigger voltage and stays on until the voltage drops below zero, even if the trigger voltage is removed. Since AC voltage is being used, it is

necessary to trigger the gate of the triac during both the negative and positive cycle of the sine wave. It is necessary to trigger the triac gate at the most minimal voltage required in order to reduce the loss of power given by the 24 VAC 60Hz power supply to the sprinkler.

The sprinkler valve control circuit is controlled using an AC switching circuit contained of a MOC3011 optically isolated triac driver chip and a 200V .8A rated triac, which can be seen in figures 12 and 13 respectively.

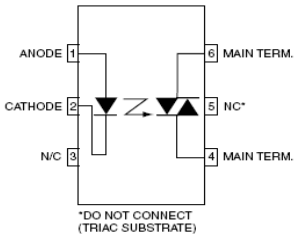
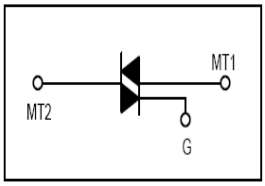
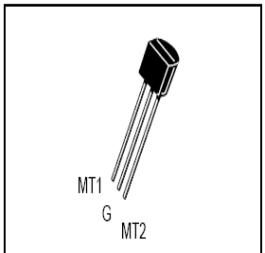


Figure 12 MOC3011 Triac Driver





CASE 29–04
TO–226AA, STYLE 12
(TO–92)

MAXIMUM RATINGS (T_J = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Peak Repetitive Off-State Voltage (Gate Open, T_J = –40 to +110°C)(1) 1/2 Sine Wave 50 to 60 Hz, Gate Open MAC97–4, MAC97A4 MAC97–6, MAC97A6 MAC97–8, MAC97A8	V_{DRM}	 200 400 600	Volts
On-State RMS Current Full Cycle Sine Wave 50 to 60 Hz (T_C = +50°C)	$I_{T(RMS)}$	0.8	Amp
Peak Non–repetitive Surge Current (One Full Cycle, 60 Hz, T_A = 110°C)	I_{TSM}	8.0	Amps
Circuit Fusing Considerations T_J = –40 to +110°C (t = 8.3 ms)	I^2t	0.26	A^2s
Peak Gate Voltage (t ≤ 2.0 μs)	V_{GM}	5.0	Volts
Peak Gate Power (t ≤ 2.0 μs)	P_{GM}	5.0	Watts
Average Gate Power (T_C = 80°C, t ≤ 8.3 ms)	$P_{G(AV)}$	0.1	Watt
Peak Gate Current (t ≤ 2.0 μs)	I_{GM}	1.0	Amp
Operating Junction Temperature Range	T_J	–40 to +110	°C
Storage Temperature Range	T_{stg}	–40 to +150	°C

Figure 13 Triac and Triac Data

Using these two elements allows the microcontroller to turn the sprinkler valve on and off easily. The MOC3011 isolates the general purpose input/output pins of the microcontroller from the 24VAC voltage needed to turn the sprinkler valves on. The MOC3011 controls the gate of the triac. By using the combination of a 330 ohm resistor and the diac, which is part of the MOC3011, successful switching control of the triac along with minimal power loss were obtained.

3.4 Graphical User Interface Subsystem

3.4.1 The Design Process of the GUI

The GUI is designed to allow users to configure and test the Automated Sprinkler System. It is able to test the moisture level of zone or moisture sensors and read the results from a microcontroller. The GUI is designed to be easy to use on a Windows based PC and with the microcontroller system.

To design the software, the overall objectives must be taken into consideration to know what must be done. These objectives are:

- To communicate with a microcontroller over Ethernet
- Allow the administrative user to edit the setup of the zones, sensors and other users.
- Perform actions in the application that cause the desired result to be performed on the microcontroller system.
- Be easy to use.

In choosing a programming language for the software application of the Automated Sprinkler System, the overall objectives must be able to be accomplished. The languages that were considered during the design process were Java, and C#. These languages were designed to allow easy GUI programming.

Since the designer didn't know much about either language C# was chosen because of a few reasons. First, the designer knew how to program in C and C++, so making the transition to C # wasn't difficult. Second, Java programming is done in a different environment than C or C# so besides learning a new language the designer would have to learn to use new development tools. Finally, the designer is very familiar with the Visual Studio .Net development tool that C# is developed in.

In the beginning communication between the microcontroller and GUI was to be done with Ethernet. After a few months of trying to connect the two with the help of the microcontroller technical support communication could not be established. The communication requirement was changed, therefore, to use a serial port, which made things easier through C#.

The communication protocols Visual Studio .Net 2005 sets up are through the serial port at a transmission rate of 57.6K baud rate. The transmission will consist of sending back and forth three to four bytes at a time with no parity bit or stop bits. In the GUI the functions for sending and receiving are:

- serialPort.Write(txBuffer, start index, number of bytes to send)
- serialPort.Read(rxBuffer, start index, number of bytes to read.)

A dialog was created to allow the users to enter which com port the microcontroller was connected to. This screen can be seen in figure 14.

Figure 14 Connection Screen

With the communication with the serial port complete, the next major objective was to create a login screen that uses an xml file format to check for user names and passwords. This was done by using an object called a dataset. The data set, keeps information in a collection in a specific format similar to a database. Along with the login screen a screen was created that allows administrators to edit users. Figure 15 and 16 show these two windows.

Figure 15 Login Screen

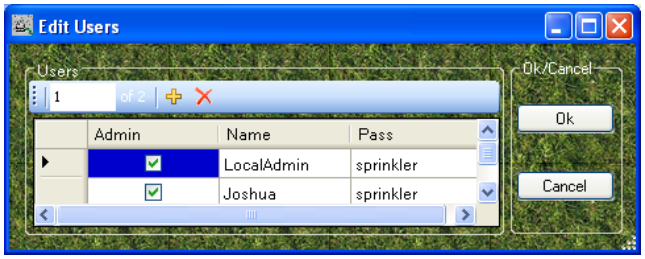


Figure 16 Edit Users Screen

Using the more datasets, the ability to add and delete sprinkler zones and moisture sensors was implemented. Once all the sensors are added to the dataset they can be sent to the microcontroller so that it knows how many zones and sensors it has connected.

Using buttons, the ability to start and stop the sprinkler system was then created. When the start button is pressed the GUI sends a three byte message “STA” to the microcontroller. When the stop button is pressed another three byte message “STO” is sent to the controller. Once the message is received by the controller it sends the same message back in response.

Finally, the objective to test a moisture level was implemented. To do this a button was created that sent a message “TZid” when a zone was selected. A message “TSidzid” is sent to the microcontroller where id is the id of the zone or sensor and zid is the zone id the sensor belongs to. Once the microcontroller tests the area it sends a response back of “TBnum” where num is the percentage of volumetric water content read from the sensor. The GUI uses this to display the level on the bar graph. Figure 17 shows the main screen.

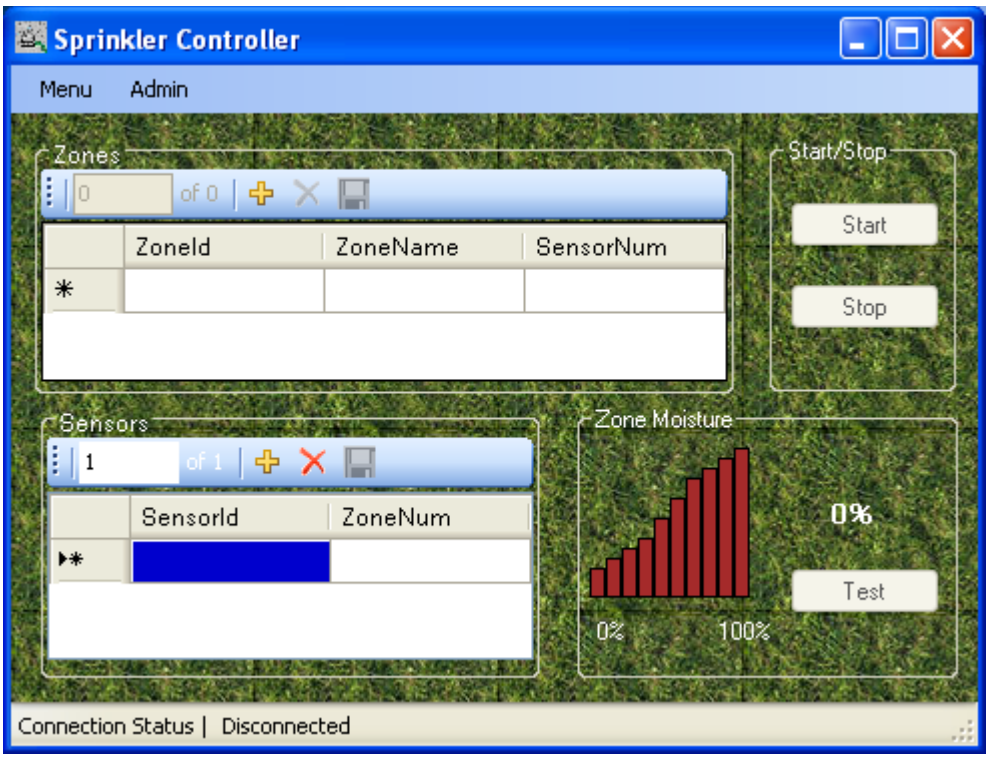


Figure 17 Main Screen

3.4.2 Graphical User Interface Flow

The flow for the GUI can is depicted in Figure 18. The GUI starts running and displays all the data that it is receiving from the microcontroller and continues to do so until an action is performed by the user. The action performs an event like those described above, and if required sends the message to the microcontroller. It then displays the change to the user and the microcontroller performs the action once decoding the received message.

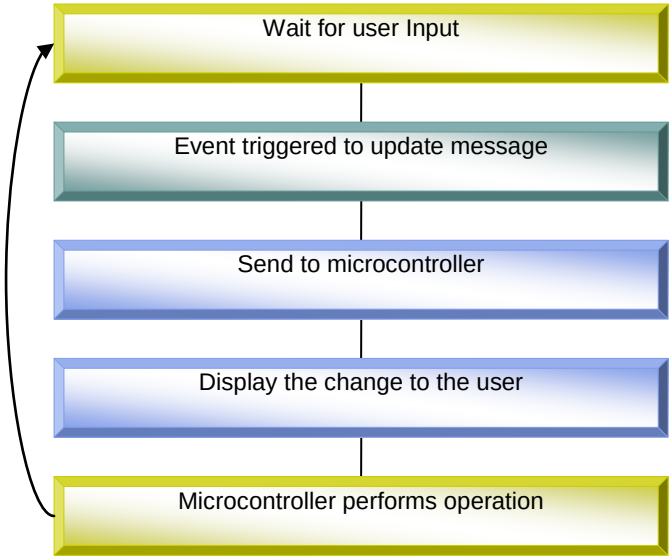

Wait for user Input
Event triggered to update message
Send to microcontroller
Display the change to the user
Microcontroller performs operation

Figure 18 GUI Flow Diagram

3.5 Microcontroller

3.5.1 Characteristics

To satisfy design requirements, a microcontroller that would be able to handle all the inputs and outputs (I/O) that the system has was needed. To meet these requirements required a microcontroller with at least twenty four I/O ports, one of which would be an analog to digital converter. There was also a need to have a microcontroller that could handle Ethernet communication to the PC. While looking for the microcontroller the designer had in mind an easy to code microcontroller that had built in function to handle the Ethernet control and other I/O controls to facilitate ease of coding. The designer was able to find all of these except for the analog to digital pin in the Zilog EZ80F91 mini microcontroller.

This Particular microcontroller has pins for up to thirty two I/Os. Along with the I/Os, there is an onboard Ethernet and UART port. The microcontroller also comes with an easy to use C code compiler that uses built in functions to control and program the Zilog. Seen in figure 19 is the pin out of the microcontroller chosen to control the Automated Sprinkler System.

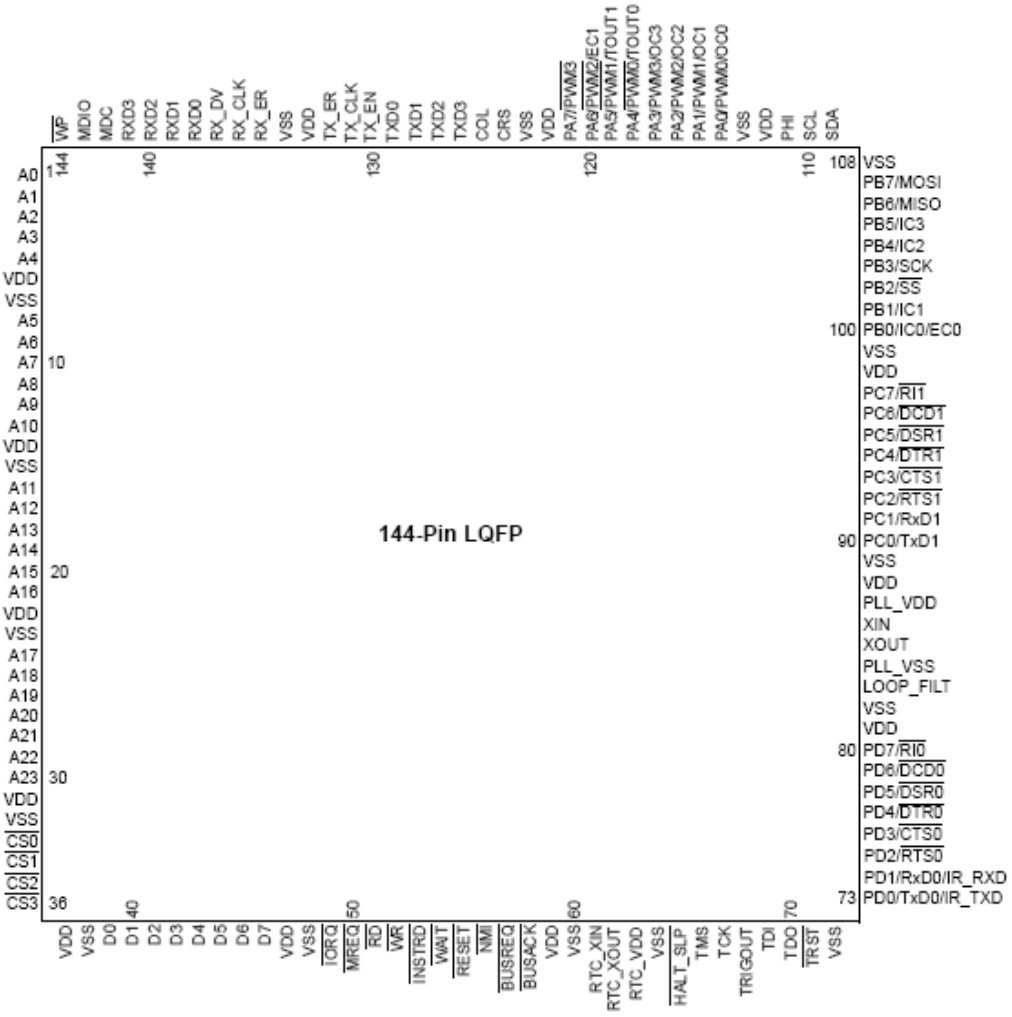


Figure 19 Zilog EZ80F91

The microcontroller comes with a modular development board. This board has two sixty pin jumpers on the bottom to allow access to the

general I/O pins and various pins for the micro processor. Figures 20 and 21 show the schematics for the development board.

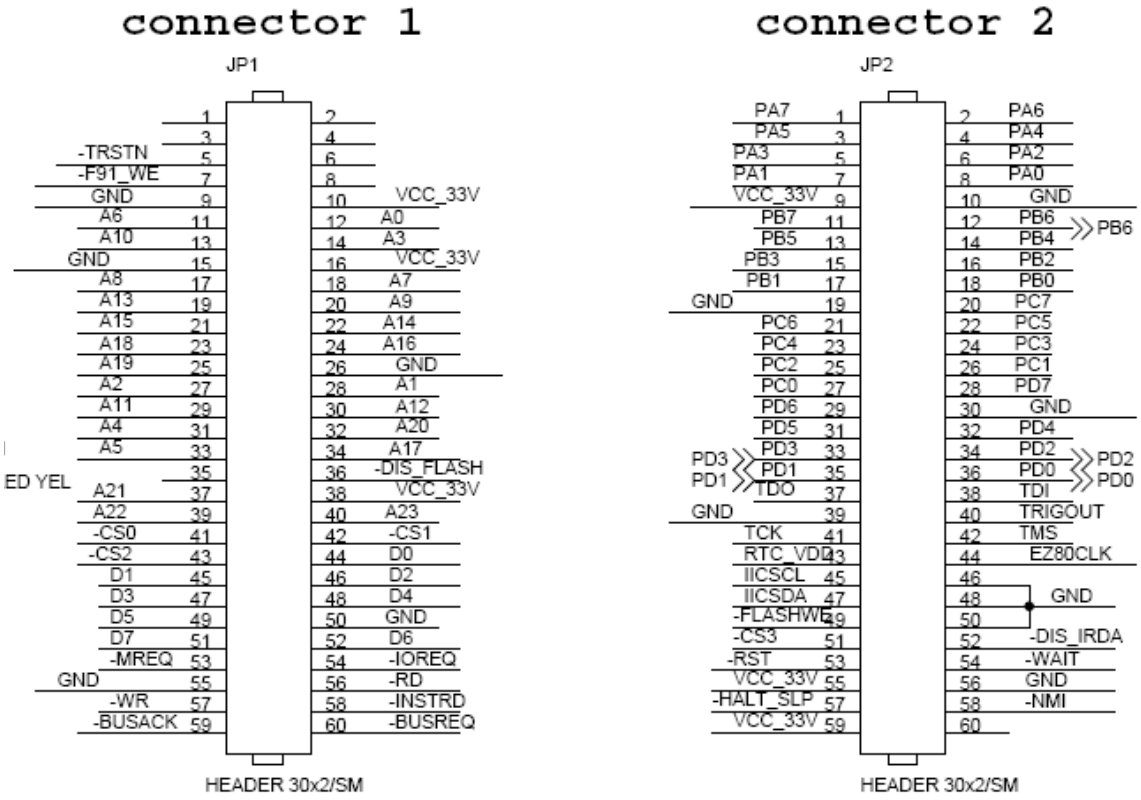


Figure 20 Development Board Header

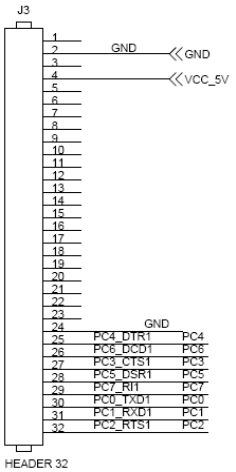


Figure 21 Development Board Header 2

3.5.2 Software Flow

An example of the setup demo software flow for the microcontroller for the system is shown in figure 22.

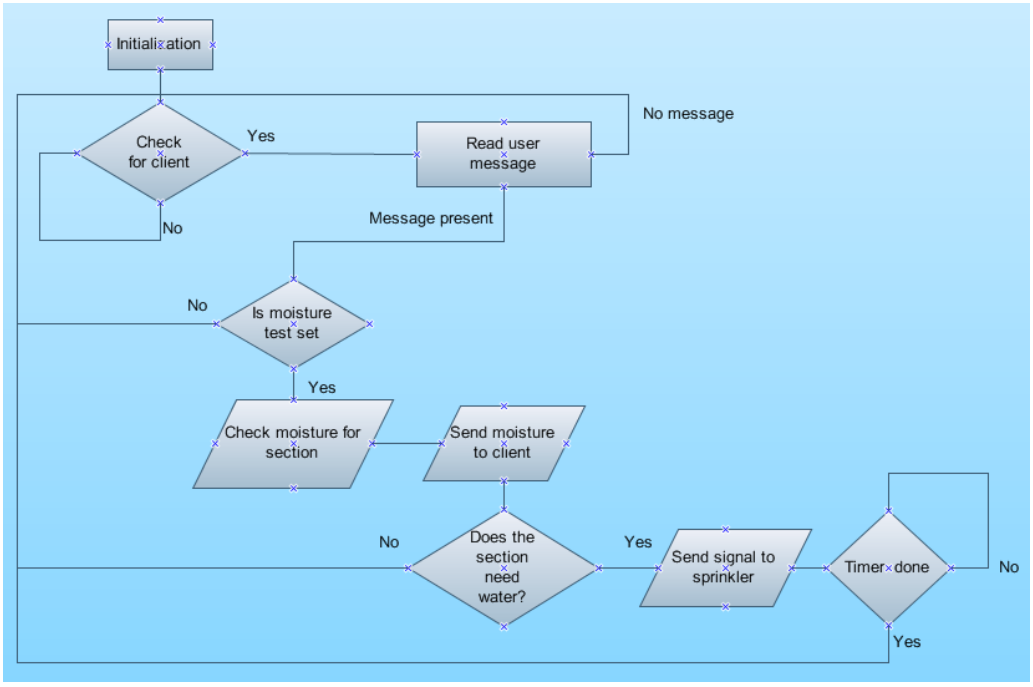


Figure 22 Software Flow

The system initializes all ports and the UART. After these have been initialized, the system waits for the start command to be sent and then sends the same command back to acknowledge it is starting. Next, the controller waits until it receives a message from the GUI. Once it does, it will decode the message and perform the task.

When a message to test a sprinkler zone is received the controller finds the chosen zone in an array of zones. It then loops through each sensor the zone has and then averages the moisture level of the sensors. Once this is finished it sends a message to the GUI with the moisture level it calculated. If the moisture level of the zone is below fifty percent

volumetric water content the microcontroller turns on the zones sprinkler valve until the water content is above fifty percent.

When a message to test a moisture sensor is received the controller finds which zone it belongs to and then reads the moisture from the sensor. When it is finished, it will send the moisture level to the GUI.

3.5.3 Demo Program

An example of the demo software flow for the microcontroller for the system is shown in figure 23.

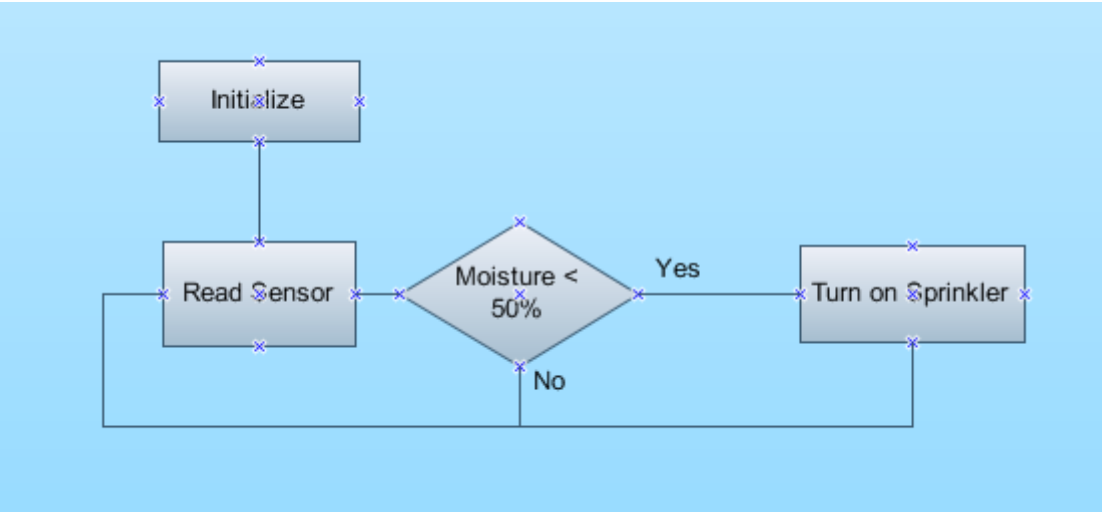


Figure 23 Demo Software Flow

The system initializes the real time operating system and all the General input/output pins. Next it will read the moisture level of the first sensor then determine if the level is too low. If it is it will turn the sprinkler valve on. Next it will then read the moisture level and keep the sprinkler valve on until the moisture level gets back above 50 %. This procedure then repeats until the controller is turned off.

3.5.4 I/O Ports

The subsystems have been assigned to different sections of the I/O ports shown in Table 1. Port A is set to be the eight bit input of the A/D converter. All of Port B is set to be an output. The first four bits of Port B is the address for the sensor analog multiplexer so the correct channel is chosen. Bit five is set to control the enabling of the sensors analog multiplexer. Bit six is designated as the enabling pin for the analog multiplexer in the sensor data circuit. Bit seven enables the A/D converter and bit eight tells the A/D converter to start converting.

The first five bits of Port C are set to be an output signal. The first three bits of the port is the address for the zone eight-to-one multiplexer. Bit four is the enable pin for the eight-to-one multiplexer and bit five is the input pin for the eight-to-one multiplexer. Bits six and seven are set to be inputs. Bit six is the ready pin from the A/D converter. This pin indicates that the converter is ready to convert a signal. Finally bit seven is the interrupt pin from the A/D converter; when it is low the conversion has been completed.

Port	Use	I or O	Pin
A	A/D Output 1	I	1
	A/D Output 2	I	2
	A/D Output 3	I	3
	A/D Output 4	I	4
	A/D Output 5	I	5
	A/D Output 6	I	6
	A/D Output 7	I	7
	A/D Output 8	I	8
B	Sensor Address 0	O	1

	Sensor Address 1	O	2
	Sensor Address 2	O	3
	Sensor Address 3	O	4
	Sensor Power En	O	5
	Sensor Data En	O	6
	A/D Chip Select	O	7
	A/D Read	O	8
C	Zone Address 0	O	1
	Zone Address 1	O	2
	Zone Address 2	O	3
	Zone Enable	O	4
	8:1 Mux Input	O	5
	A/D Ready	I	6
	A/D Int	I	7
			8

Table 1 I/O Ports

4. Budget

The budget for the project is not specific, but minimization of cost is greatly desired. The estimated budget was expected to be about two hundred dollars. The needed software for microcontroller software development was included in the development board package, and software for the GUI was downloaded from www.ece.usu.edu/msdnaa.php as a release of Visual Studio .Net 2005. The expected costs will be for the actual hardware. Table 2 shows the overall cost of the project.

Description	Part Number	Manufacture	Vendor	Quantity	Cost	Total
Micro Controller	eZ80F910200KITG	Zilog	Zilog	1	$94.50	$94.50
Moisture Sensor	EC 5	Echo	Decagon Devices	1	$109	$109
Optical Isolator	MOC3011	FairChild	Fairchild Direct	1	$0.67	$0.67
Sprinkler Valve	57041	Orbit WaterMaster	Home Depot	1	$7.97	$7.97
A/D Converter	ADC0320	Maxim IC	Maxim Ic	1	Free	Free
24 VAC	57040	Orbit WaterMaster	Home Depot	1	$14.27	$14.27
16:1 Mux	Max396	Maxim IC	Maxim Ic	2	Free	Free
8:01 Mux	Max358	Maxim IC	Maxim Ic	1	Free	Free

Triac	Mac97A4	FairChild	Fairchild Direct	1	$1.12	$1.12
Misc Circuit	Misc	Misc	Ece Store	1	$29.92	$29.92
					Total	$257.45

Table 2 Project Cost

5. Project Management

5.1 Schedule Tasks

1. Proposal
2. System Design
    - 2.1 Functional/Block diagram of hardware and software
    - 2.2 GUI layout and options
3. Obtain parts
4. Learn how to use hardware
5. Design
    - 5.1 Software programs (GUI and microcontroller)
    - 5.2 Hardware schematics
6. GUI
    - 6.1 Compatibility with Ethernet and microcontroller
    - 6.2 Final layout
7. Assembly
8. Testing

These tasks reflect my desired results and the time it should take to accomplish the above tasks. The start date for the project is August 28, 2007 and the projected finishing date is April 28th, 2008. The schedule of the tasks can be seen in Table 3. Each column represents a week's time.

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
2.1															
2.2															
3															
4															
5															
5.1															
5.2															
6															
6.1															
6.2															
7															
8															

Table 3 Gantt chart

6. Conclusion

While maintaining a relatively low budget of around two hundred thirty dollars, the Automated Sprinkler system will provide a more efficient sprinkler system. The package will include moisture sensors, the microcontroller and a software package. The sensors are used to measure the moisture of the soil for certain zones in the sprinkler system. Applications I have learned from school and from work enabled me to complete all the tasks outlined in the schedule of the Gantt chart and finish the project on the 27th of April, 2008.